HiPERiSM Consulting, LLC.

George Delic , Ph.D.

HiPERiSM Consulting, LLC

(919)484-9803

P.O. Box 569,

Chapel Hill, NC 27514

george@hiperism.com

http://www.hiperism.com

U.S. EPA 9th Modeling Conference
October 9-10, 2008
EPA RTP Campus, NC

REGULATORY AIR QUALITY MODELS
FOR
NEXT GENERATION COMPUTERS:
PROSPECTS AND CHALLENGES

George Delic, HiPERiSM Consulting, LLC

HiPERiSM

Overview

1. Introduction
2. Identifying the problem
3. Computer hardware
4. Examples of AQM performance
5. U.S. EPA AQM models: lessons learned
6. Can software and hardware help?
7. Next steps
8. Outcomes
9. Disclaimer

1. Introduction

- Regulatory Air Quality Models (AQM)
  - ➢Developed by the U.S. EPA (and contractors)
  - ➢Their use is mandatory for SIPs
  - ➢They require ***long*** model runs
  - ➢They have a dedicated user community forced to invest in support infrastructure: software, hardware, HR staff
- Hardware and programming environment
  - ➢Revolutionary developments are here now!
  - ➢Other modeling disciplines report cost benefit enhancements of 50 to 100 times more

2. Identifying the problem

- Performance
  - ➢ HiPERiSM's investigations with such models shows:
    - ✓ Many inefficiencies with mediocre to poor performance
    - ✓ Mismatch to current commodity-of-the-shelf (COTS) hardware
    - ✓ Worse performance on next generation computers
- The situation for AQM's
  - ➢ The AQM community needs help and leadership
  - ➢ Does the U.S. EPA have a plan to face the challenge for change in COTS hardware?

2. Identifying the problem (cont.)

Question	Answer
What is the problem?	Movement of data is now considered to be the single most expensive operation on commodity platforms
Don't modern architectures solve the problem?	They do this by inserting complex memory hierarchies, but this challenges an application's ability to extract optimal performance from commodity solutions
What can be done to fix the problem?	Fully understand the memory architecture's impact on application performance and then fix the problem at the source

Multicore processors exacerbate the problem because:

- concurrently executing threads compete for memory bandwidth
- the effective cache size per thread is diminished

3. Computer Hardware

Current generation: multicore	**Next generation: *many*core**
2-4 cores per CPU	8 – 100's cores per CPU
Cache Level 1, 2, or 3	Level 1 for each core and Level 2 shared across cores
CPUs access memory via bus	Cores access subset of L2 and memory via bus

The GPGPU revolution: Multi-processing graphics hardware on outboard processors with programming tools for hundreds of parallel threads.

3. Computer Hardware (cont.)

Memory and cache

- The memory hierarchy uses cache to hide the negative effects of memory latency
- Cache space is wasted when data resides there but is unused
- Unused data in cache consumed precious bandwidth when it was loaded from memory

4. Examples of AQM performance

- SOM an Ocean Model: example (a)
  - Used as a reference
- CMAQ: examples (b) and (c)
  - Rosenbrock solver (ROS3)
  - Euler Backward solver (EBI)
- AERMOD: example (d)

All the above models used these HiPERiSM resources:

• A 64-bit (x86_64) Linux platform with a 16KB L1 data cache and 1MB L2 cache with compilers typically used by the U.S. EPA (using EPA code for CMAQ and AERMOD)

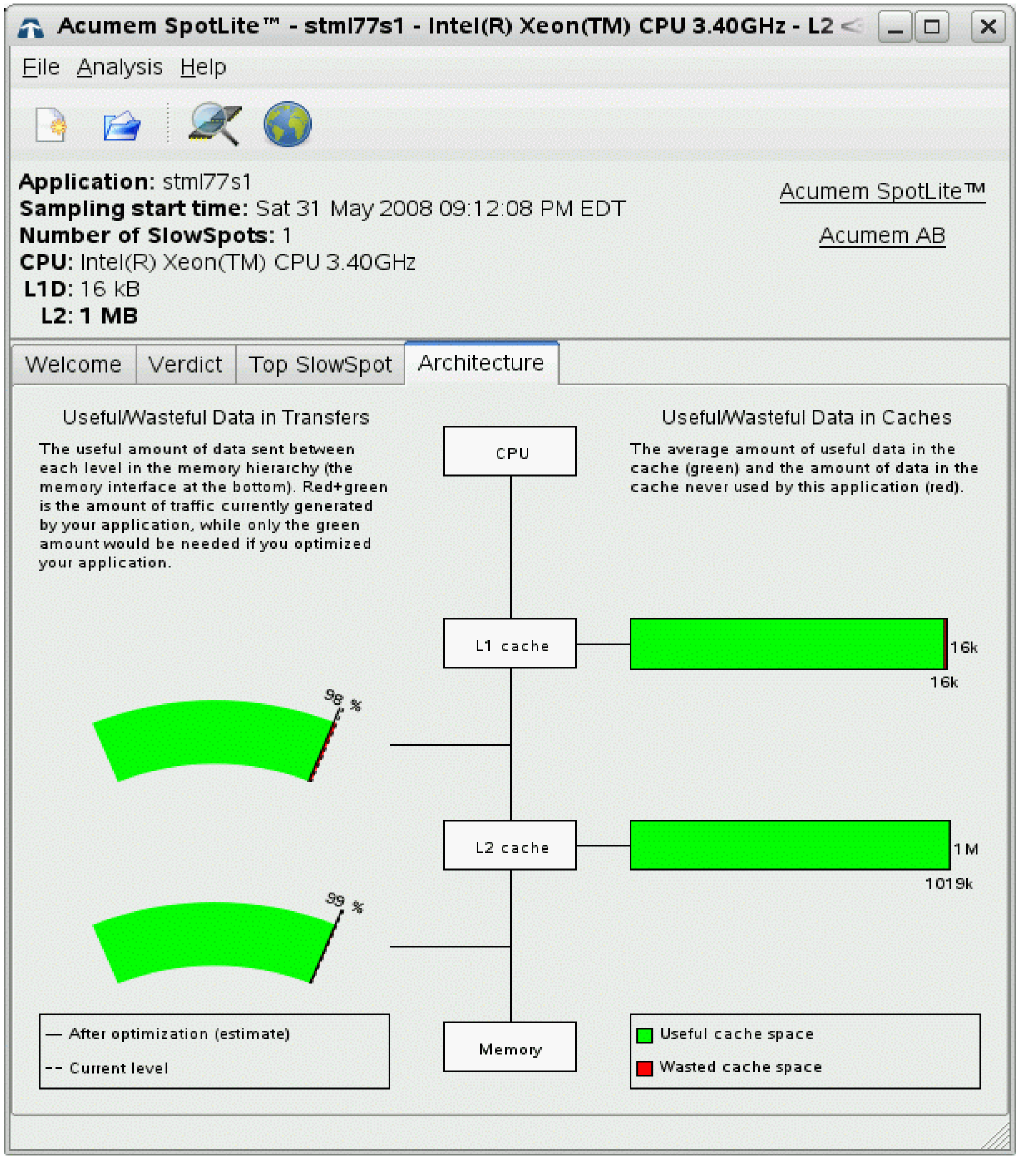
• SlowSpotter™ software from Acumem®, Inc. to collect performance data (for details see HiPERiSM's Web URL)

4. Example (a) SOM Ocean Model

Excellent cache utilization

GREEN on the right hand-side bars shows no wasted cache space – i.e. *all data loaded from memory is used by the CPU*

(Single CPU with one core and two cache levels)



 http://www.hiperism.com

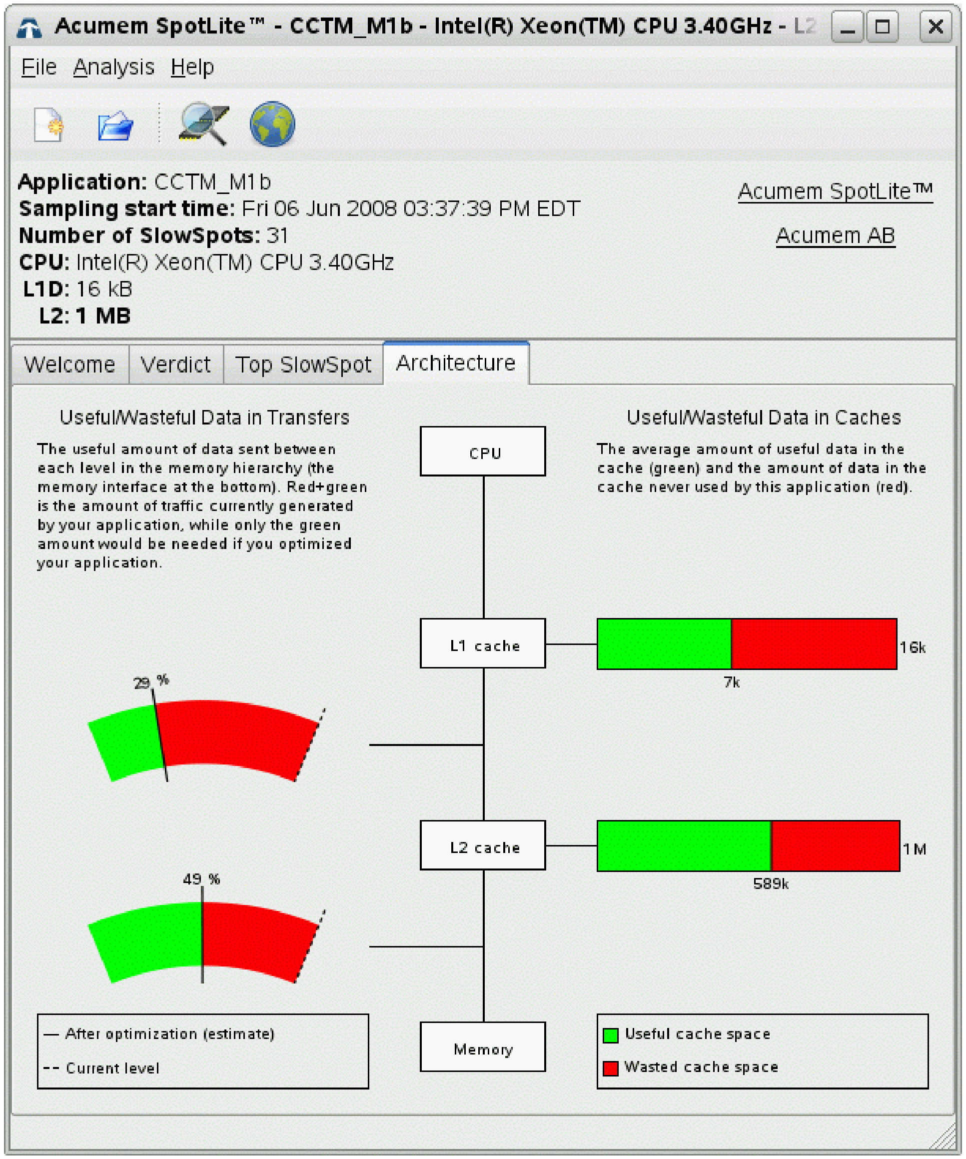
HiPERiSM

4. Example (b) CMAQ ROS3 Solver

Mediocre cache utilization:

RED on the right hand-side bars shows wasted cache space – i.e. *data loaded from memory but never used*

(Single CPU with one core and two cache levels)



 http://www.hiperism.com

4. Example (c) CMAQ EBI Solver

Comparing CMAQ solvers* (EBI versus ROS3):

- EBI: 3x *more wasted cache space*
- EBI: 4x *worse memory prefetching performance*

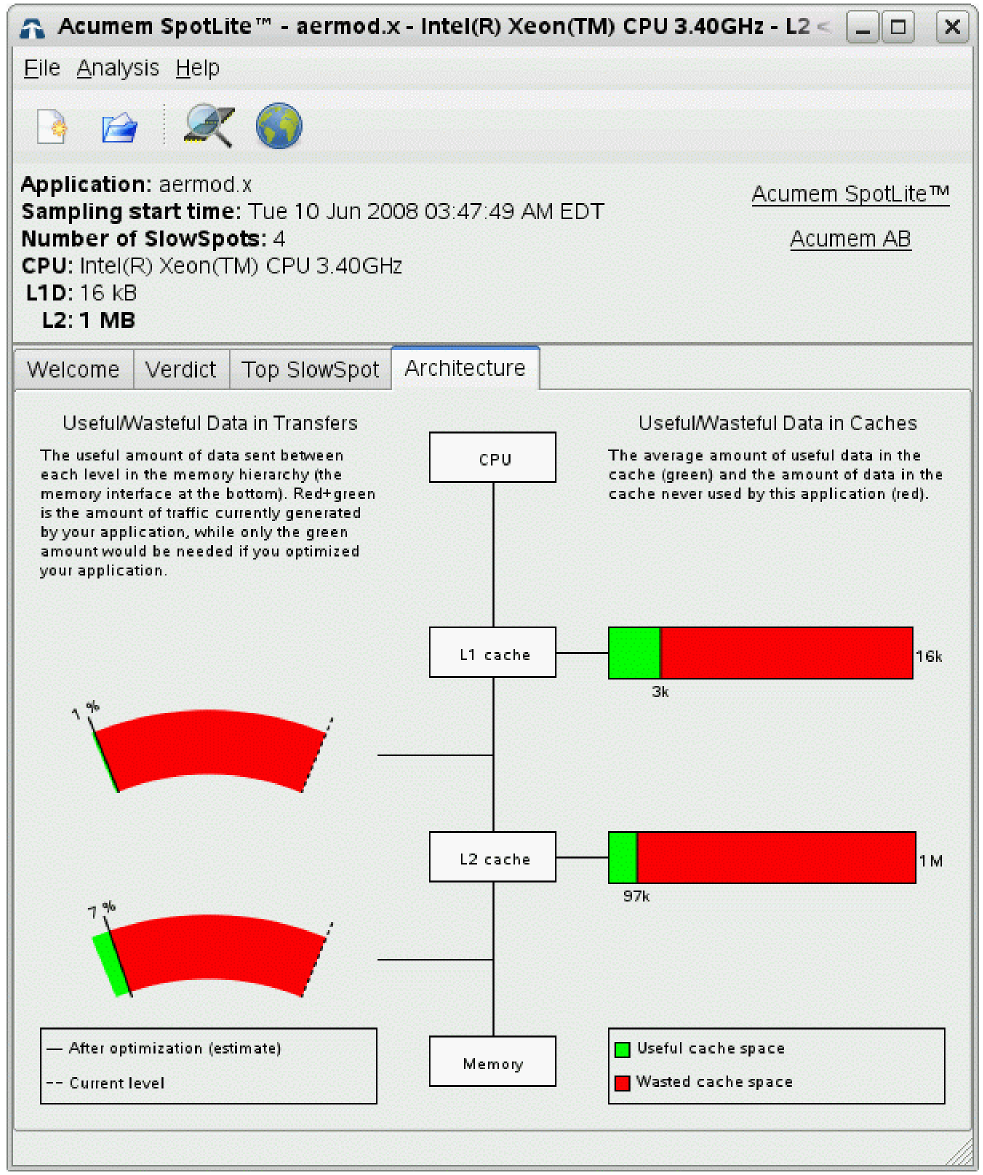
* Linux platform with a 16KB L1 data cache and 1MB L2 cache for the mid-morning hours of a summer episode (14 August, 2006)

4. Example (d) AERMOD

Poor cache utilization:

RED on the right hand-side bars shows wasted cache space – i.e. *data loaded from memory but never used*

(Single CPU with one core and two cache levels)



 http://www.hiperism.com

5. U.S. EPA's AQM models: lessons learned

Memory footprint of AQM's:

- Inherent in the current state of models:
  - inefficient use of COTS hardware
  - lost performance opportunities
- Critical bottle-necks in memory access:
  - cache utilization is wasteful
  - cost of latency leads to CPU stalls

6. Can software or hardware help?

Compilers will not solve the performance bottle-necks because:

- The code lacks the right structure
- Requires too much disorganized data movement

Next generation hardware requires data parallelism:

- Needs to be expressed in the code by the developer
- It cannot be discovered by compilers

7. Next Steps

U.S. EPA needs to show leadership by:

- Soliciting input from the community
- Developing an action plan to meet the challenge
- Provide resources for change
    - ❑ ***Consequences of inaction include:***
        - ➢ ***lowered performance, and***
        - ➢ ***escalating support infrastructure costs***

HiPERiSM

8. Outcomes

- ***GREEN COMPUTING !***
  - ***More efficient use of COTS computers***
- ***Lower cost of AQM support infrastructure***
  - ***Higher throughput = fewer resources required***
- Cost benefit analysis suggests:
  - Modification of AQM's will yield
    - boost in throughput by orders of magnitude
    - lower TCO (total cost of ownership)

9. Disclaimer

None of the work reported here has been sponsored or funded by the U.S. EPA

Further information is available at:

http://www.hiperism.com

http://www.hiclas1.com

